## WHAT IS CLAIMED IS:

1		1.	An application programming interface comprising:
2		a first	interface which controls transfer of information between a first
3	device capabl	e of har	adling isochronous and asynchronous data and an audio/video file
4	system capabl	le of hai	ndling and organizing audio/video data.
		2	The application are comming intenfers according to claim 1
1	G41	2.	The application programming interface according to claim 1,
2	further compr	•	
3			nd interface which controls transfer of information between a second
4	device capabl	e of har	ndling asynchronous data and said audio/video file system.
1		3.	The application programming interface according to claim 1,
2	wherein said	first dev	rice is an audio/video controller.
1 .		4.	The application programming interface according to claim 3,
2	wherein said		ideo controller is capable of processing commands transmitted using
			dec controller is capable of processing commands transmitted using
3	protocol 6188	3.	
1		5.	The application programming interface according to claim 4,
2	wherein said	commai	nds are transmitted using protocol 61883 in an isochronous manner.
1		6.	The application programming interface according to claim 2,
2	wherein said		device is a SBP-2 controller.
-	Wildian Sara	occorra v	
1		7.	The application programming interface according to claim 6,
2	wherein said	SBP-2 c	controller is capable of processing commands transmitted using
3	serial-bus-pro	tocol-2	•
1		8.	The application programming interface according to claim 7,
2	wherein said	commai	nds are transmitted using serial-bus-protocol-2 in an asynchronous
3	manner.		
1		9.	The application programming interface according to claim 1,
2	wherein contr	ol of sa	id transfer of information to and from said first device are
3	independent o	of intern	al implementation of said first device.

1	10. The application programming interface according to claim 2,
2	wherein control of said transfer of information to and from said second device are
3	independent of internal implementation of said second device.
1	11. The application programming interface according to claim 1 further
2	comprising:  a plurality of function calls.
J	
1	12. The application programming interface according to claim 11,
2	wherein one or more of said plurality of function calls are designed to allow said
3	audio/video file system to handle a first type of file; and wherein one or more of said
4	plurality of function calls are designed to allow said audio/video file system to handle a
5	second type of file.
1	13. The application programming interface according to claim 12,
2	wherein said first type of file is a non-audio/video file; and wherein said second type of
3	file is an audio/video file.
_	at the state of th
1	14. The application programming interface according to claim 12,
2	wherein said first type of file is smaller than said second type of file.
1	15. The application programming interface according to claim 11,
2	wherein one or more of said plurality of function calls are designed to allow said
3	audio/video file system to play or record a plurality of audio/video data streams
4	concurrently.
1	16. The application programming interface according to claim 15,
2	wherein said one or more of said plurality of function calls are designed to allow said
3	audio/video file system to play or record said plurality of audio/video data streams
	concurrently by using a channel ID parameter and an object ID parameter.
4	concurrently by using a channel in parameter and an object in parameter.
1	17. The application programming interface according to claim 11,
2	wherein one or more of said plurality of function calls are designed to allow said
3	audio/video file system to play and record an audio/video data stream concurrently.

18. The application programming interface according to claim 17,
wherein said one or more of said plurality of function calls are designed to allow said
audio/video file system to play and record said audio/video data stream concurrently by
using a channel ID parameter and an object ID parameter.

- 19. The application programming interface according to claim 11, wherein one or more of said plurality of function calls are designed to allow said audio/video file system to initiate a play or record operation starting from within an audio/video file.
- 20. The application programming interface according to claim 19, wherein said one or more of said plurality of function calls are designed to allow said audio/video file system to initiate a play or record operation starting from within said audio/video file by using an offset parameter.
- 21. The application programming interface according to claim 11, wherein one or more said plurality of function calls are designed to allow said audio/video file system to optimize disk access.
- 22. The application programming interface according to claim 21, wherein said one or more of said plurality of function calls are designed to allow said audio/video file system to optimize disk access by designating a first group of function calls to handle a first type of file and a second group of function calls to handle a second type of file.
- 23. The application programming interface according to claim 22, wherein said first type of file is a non-audio/video file; and wherein said second type of file is an audio/video file.
- 24. The application programming interface according to claim 11, wherein one or more of said plurality of function calls are designed to allow said audio/video file system to perform a plurality of trick operations with a data stream.
- The application programming interface according to claim 24, wherein said plurality of trick operations includes a plurality of forward operations.

1	26.	The application programming interface according to claim 25,
2	wherein said plural	ity of forward operations includes a fast-forward operation, a slow-
3	forward operation,	and a step-forward operation.
1	27.	The application programming interface according to claim 24,
2	wherein said plural	ity of trick operations includes a plurality of reverse operations.
1	28.	The application programming interface according to claim 27,
2	wherein said plural	ity of reverse operations includes a fast-reverse operation, a slow-
3	reverse operation,	and a step-reverse operation.
1	29.	An application programming interface for providing an interface
2	with an audio/vide	o file system capable of handling and organizing audio/video data,
3	comprising:	
4	a fir	est plurality of function calls including:
5		a load function call designed to cause retrieval of descriptor
6	information from a	storage medium;
7		a store function call designed to cause storing of said descriptor
8	information onto s	aid storage medium;
9		a delete function call designed to cause deletion of said descriptor
10	information from s	aid storage medium; and
11	a se	cond plurality of function calls including:
12		a play function call designed to cause a specified file to be played;
13		a record function call designed to cause specified data to be
14	recorded; and	
15		a stop function call designed to cause a play or record operation to
16	be stopped.	
1	30.	The application programming interface according to claim 29,
2	wherein said first p	plurality of function calls is designed to handle a first type of file; and
3	wherein said secon	nd plurality of function calls is designed to handle a second type of file
1	31.	The application programming interface according to claim 30,
2	wherein said first t	type of file is a non-audio/video file; and wherein said second type of
3	file is an audio/vid	eo file.

1	32.	The application programming interface according to claim 29,
2	wherein said first	plurality of function calls further includes:
3		a validity function call designed to verify validity of a specified
4	descriptor; and	
5	wherein said seco	nd plurality of function calls further includes:
6	•	a pause function call designed to cause a play or record operation
7	to be paused;	
8		a resume function call designed to cause a previously paused
9	operation to resur	ne; and
10		an address retrieval function call designed to determine a logical
11	block address of s	aid specified file during a play or a record operation.
1	33	The application programming interface according to claim 29,
1 2		nd plurality of function calls includes:
3		plurality of function calls designed to cause forward operations to be
4	performed; and	further of function cans designed to cause 150 mass of
5	•	olurality of function calls designed to cause reverse operations to be
6	performed.	Turion, or rune of the control of th
J	portormou	
1	. 34	
2	wherein said plur	ality of function calls designed to cause forward operations to be
3	performed includ	es:
4	a i	ast-forward function call;
5	as	slow-forward function call; and
6	as	step-forward function call.
1	35	The application programming interface according to claim 33,
1 2		ality of function calls designed to cause reverse operations to be
3	performed includ	
<i>3</i>	•	fast-reverse function call;
		slow-reverse function call; and
5		step-reverse function call.
6	a	step-reverse function can.
1	36	The application programming interface according to claim 29,
2	wherein said app	lication programming interface is capable of being used by a first device

2

3	capable of hand audio/video file		ochronous and asynchronous data to communicate with said n.
1 2			The application programming interface according to claim 36, ce is an audio/video controller.
1 2 3 4	wherein said ap	plication	The application programming interface according to claim 36, on programming interface is capable of being used by a second dling asynchronous data to communicate with said audio/video file
1 2		39. rst devi	The application programming interface according to claim 38; ice is a SBP-2 controller.
1 2		40. ecified	The application programming interface according to claim 32, descriptor is an object descriptor.
1 2		41. ecified	The application programming interface according to claim 32, descriptor is a content list.
1 2		42. secified	The application programming interface according to claim 32, descriptor is a performance list.
1 2		43. pecified	The application programming interface according to claim 32, d descriptor is a HMS table.
1 2 3			The application programming interface according to claim 32, first and second plurality of function calls is capable of passing a rs.
1 2 3 4	call includes a	descrij	The application programming interface according to claim 44, of parameters that is capable of being passed by said load function ptor ID parameter, a type parameter, an offset parameter, a size ration parameter, and a call_back parameter.
1		46.	The application programming interface according to claim 44,

wherein said plurality of parameters that is capable of being passed by said store function

3	call includes a descriptor ID parameter, a type parameter, an offset parameter, a size
4	parameter, a data_location parameter, and a call_back parameter.
1	47. The application programming interface according to claim 44,
2	wherein said plurality of parameters that is capable of being passed by said delete
3	function call includes a descriptor ID parameter, a type parameter, and a call_back
4	parameter.
1	48. The application programming interface according to claim 44,
2	wherein said plurality of parameters that is capable of being passed by said play function
3	call includes a channel ID parameter, an object ID parameter, a start_position parameter,
4	an_end position parameter, a speed parameter, and a call_back parameter.
1	49. The application programming interface according to claim 44,
2	wherein said plurality of parameters that are capable of being passed by said record
3	function call include a channel ID parameter, an object ID parameter, a start_position
4	parameter, a type parameter, and a call_back parameter.
1	` The application programming interface according to claim 44,
2	wherein said plurality of parameters that is capable of being passed by said stop function
3	call includes a channel ID parameter, a call_back parameter, and a logical_byte_address
4	parameter.
1	51. The application programming interface according to claim 44,
2	wherein said plurality of parameters that is capable of being passed by said pause function
3	call includes a channel ID parameter, a call_back parameter, and a logical_byte_address
4	parameter.
1	52. The application programming interface according to claim 44,
2	wherein said plurality of parameters that is capable of being passed by said resume
3	function call includes a channel ID parameter and a call_back parameter.

53. The application programming interface according to claim 44, wherein said plurality of parameters that is capable of being passed by said address retrieval function call includes a channel ID parameter and a count parameter.

1	54. Th	e application programming interface according to claim 44,
2	wherein said plurality of	parameters that is capable of being passed by said validity
3	function call includes a d	escriptor ID parameter, a type parameter and a call_back
4	parameter.	
1	55. Th	e application programming interface according to claim 34,
2	wherein said fast-forward	I function call is capable of passing a plurality of parameters
3	including a channel ID pa	arameter, a type parameter, an interval parameter, a repeat
4	parameter, and a call_bac	ek parameter.
1	56. Th	e application programming interface according to claim 34,
2	wherein said slow-forward	ed function call is capable of passing a plurality of parameters
3	including a channel ID pa	arameter, a repeat parameter, an increment parameter and a
4	call_back parameter.	
1	57. Th	e application programming interface according to claim 34,
2	wherein said step-forwar	d function call is capable of passing a plurality of parameters
3		arameter, an increment parameter and a call_back parameter.
1	58. Th	e application programming interface according to claim 35,
2		function call is capable of passing a plurality of parameters
3		arameter, a type parameter, an interval parameter, a repeat
4	parameter, and a call_bac	
	-	
1		ne application programming interface according to claim 35,
2		e function call is capable of passing a plurality of parameters
3		arameter, a repeat parameter, an increment parameter and a
4	call_back parameter.	·
1	60. Th	ne application programming interface according to claim 35,
2	wherein said step-reverse	e function call is capable of passing a plurality of parameters
3	including a channel ID p	arameter, an increment parameter and a call_back parameter.
1	61. A	method for providing communication with an audio/video file
2	system, comprising steps	s of:

	3	providing a first interface which controls transfers of information between
	4	said audio/video system and a first device capable of handling isochronous and
	5	asynchronous data; and
	6	providing a second interface which controls transfers of information
	7	between said audio/video system and a second device capable of handling asynchronous
	8	data.
1		62. The method according to claim 61, wherein said signals transferred
2	be	tween said audio/video system and said first device are independent of internal
3	im	plementation of said first device; and
4		wherein said signals transferred between said audio/video system and said
5	se	cond device are independent of internal implementation of said second device.